Subspace-based Direct Visual Servoing

Eric Marchand

Subspace-based Direct Visual Servoing

Eric Marchand

***Abstract*—To date most of visual servoing approaches have relied on the geometric features that have to be tracked and matched in the image. Recent works have highlighted the importance of taking into account the photometric information of the entire images. This leads to direct visual servoing (DVS) approaches. The main disadvantage of DVS is its small convergence domain compared to conventional techniques, which is due to the high non-linearities of the cost function to be minimized. In this paper we propose to project the image on an orthogonal basis (PCA) and then servo on either images reconstructed from this new compact set of coordinates or directly on these coordinates used as visual features. In both cases we exhibit the analytical formulation of the interaction matrix. We show that these approaches feature a better behavior that the classical photometric visual servoing scheme allowing larger displacements and a satisfactory decrease of the error norm thanks to a well modelled interaction matrix.**

***Index Terms*—Visual servoing, sensor-based control**

I. Introduction

VISUAL servoing uses the information provided by a vision sensor to control the movements of a dynamic system [6]. This approach requires the extraction of visual information (usually geometric features) from the image in order to design the control law. Robust extraction and real-time spatio-temporal tracking of these visual cues is to date a non trivial task.

While there has been progress in extracting and tracking relevant features, a new approach called direct visual servoing (DVS) has been emerging for almost 10 years now [15], [8], [7], [10]. It has been demonstrated that only the luminous intensities of the images can be taken into account to control the robot and that conventional tracking processes can be avoided. The main drawback of DVS is its convergence domain compared to classical VS techniques, which is due to the high non-linearities of the cost function to be minimized. Various schemes have been proposed in order to improve the robustness of DVS by considering various descriptors (image intensity, gradient, color, etc.) or cost functions (mutual information [10], histogram distances [2], mixture of Gaussians [9]). Recently, it has been proposed to consider convolutional neural network to bypass the modelling step [3].

Another solution to increase the convergence domain would be extract from the image a set of coefficients that could then be used as control input. The idea is not to extract geometric features from the image (as it is usually done in visual servoing) but to "compress" the original image information in order to get a compact representation. This is what has been done with photometric moments which allows retain geometric information [1]. It was shown that it provides a better behavior that a classical control based on points [6] and extends significantly the convergence of the photometric visual servoing approach [7]. Another interesting approach was proposed in order to consider Wavelet [19], [13] and Shearlet-based [13] image representations and thus to consider in the control law the wavelet/shearlet coefficients. The authors [19], [13] then proposed an analytical formulation of the interaction matrix that links the variation of these coefficients to the camera motion.

Actually, one of the very first attempt to achieved DVS has been proposed in [18], [12], [11]. In these papers, image intensity is not directly considered but an eigenspace decomposition is performed to reduce the dimensionality of image data. This is done thanks to a Principal Component Analysis (PCA) process (also known as Karhunen-Loève expansion). The control is then performed on the image coordinates in the eigenspace (an orthogonal basis). This process requires the off-line computation of this eigenspace and then, for each new frame, the projection of the image on this subspace in order to compute the set of coordinates (coefficients) in the new basis that will be used in the control law. An interest of such approach is that, when projecting an incoming image on this basis, the greatest variance comes to lie on the first coefficient, the second greatest variance lies on the second coefficient, etc. Only a few coefficients allow to grasp most of the variance of the image. Nevertheless, in [18], [12], [11], the interaction matrix related to the eigenspace is not computed analytically but is estimated on-line from the estimation of the plane coordinates tangent to the cost function surface leading to unsatisfactory behavior. In [11] the author points out the importance of using proper interaction matrices for visual servoing.

In this article, we clearly draw inspiration from Deguchi's previous work [11]. Based on our own previous work on DVS [7], we propose two new control laws based on a principal analysis decomposition of the main component of the image. In both case we first project the image on the new orthogonal basis and obtain a new and compact set of coordinates (coefficients). The former approach is a photometric visual servoing technique that considers, as input, an image reconstructed from a small number of coefficients. Indeed, an approximation of the image can be obtained as a linear combination of a small subset of coefficients and the eigenspace. Within this context, we remain in the photometric visual servoing scheme and provide an analytical formulation of the interaction matrix. Dealing with the latter approach, we consider the coordinates in the new basis as the visual

Manuscript received February, 23, 2019; Revised April, 29, 2019; Accepted April, 30, 2019.

This paper was recommended for publication by Editor C. Cadena upon evaluation of the Associate Editor and Reviewers' comments.

Eric Marchand is with Univ Rennes, Inria CNRS, IRISA, Rennes, France e-mail: Eric.Marchand@irisa.fr.

Digital Object Identifier (DOI): see top of this page.

features. Again, within this context we propose an explicit and analytical formulation of the interaction matrix. We show on various experiments including real 6 DoF positioning tasks, that these approaches feature a better behavior that the classical photometric visual servoing scheme allowing larger displacements and a satisfactory decrease of the error norm thanks to a well modelled interaction matrix.

In the reminder of this paper, Section II gives an overview of the DVS scheme. Section III recalls the principal component analysis concept and the way to build the eigenspace and obtain the new image coordinates. Section IV gives the details of the control laws including the derivation of the related interaction matrix. Finally, Section V illustrates the effectiveness of the approach: first with simulated results but also with experiments carried out on a 6 DoF Viper 850 robot.

II. Direct Visual Servoing

A. Positioning Task by visual servoing

The aim of a positioning task is to reach a desired pose of the camera $\mathbf{r}^*$, starting from an arbitrary initial pose. To achieve that goal, one needs to define a cost function that reflects, in the image space, this error. Most of the time this cost function is an error measure which needs to be minimized. Considering the actual pose of the camera $\mathbf{r}$ the problem can therefore be written as an optimization process:

$$\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}} \mathbf{e}(\mathbf{r}). \tag{1}$$

For example, considering a set of geometrical features $\mathbf{s}$, the task will typically have to minimize the error $\mathbf{e}(\mathbf{r})$ that is the difference between the current $\mathbf{s}(\mathbf{r})$ and the desired configuration $\mathbf{s}^*$, classically $\mathbf{e}(\mathbf{r}) = \mathbf{s}(\mathbf{r}) - \mathbf{s}^*$.

This visual servoing task is achieved by iteratively applying a velocity to the camera. This requires the knowledge of the interaction matrix $\mathbf{L_s}$ of $\mathbf{s}(\mathbf{r})$ that links the variation of $\dot{\mathbf{s}}$ to the camera velocity and which is defined as [14], [6]:

$$\dot{\mathbf{s}}(\mathbf{r}) = \mathbf{L_s}\mathbf{v} \tag{2}$$

where $\mathbf{v}$ is the camera velocity.

This equation leads to the expression of the velocity that needs to be applied to the robot. The control law is classically given by [6]:

$$\mathbf{v} = -\lambda \mathbf{L_s}^{+}\mathbf{e}(\mathbf{r}) \tag{3}$$

where λ is a positive scalar and $\mathbf{L_s}^{+}$ is the pseudo inverse of $\mathbf{L_s}$.

This approach requires to extract and track visual information (usually geometric features) from the image in order to design the control law. This difficult tracking process is one of the bottlenecks in the development of visual servoing techniques.

B. Photometric visual servoing

Recent works have tried to circumvent these problems by using directly the information provided by the entire image [8]. Features are no longer extracted from the image. In [7], a control law was proposed that minimizes the error between the current image and the desired one. In that case the vector of visual feature in nothing but the image itself and the error to be regulated is the sum of squared differences (the SSD).

In that case, the feature $\mathbf{s}$ becomes the image itself ($\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r})$). This means that the optimization process becomes [7]:

$$\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}} (\mathbf{I}(\mathbf{r}) - \mathbf{I}^*) \tag{4}$$

where $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$ are respectively the image seen at the position $\mathbf{r}$ and the template image (both of N pixels). The control law is inspired by the Levenberg-Marquardt (LM) optimization approach. It is given by:

$$\mathbf{v} = -\lambda \mathbf{L_I}^{+}(\mathbf{I}(\mathbf{r}) - \mathbf{I}^*) \tag{5}$$

where λ is a positive scalars and $\mathbf{L_I}$ is the interaction matrix related to the luminance [8]. If we introduce the interaction matrices $\mathbf{L}_x$ and $\mathbf{L}_y$ related to the coordinates x and y of $\mathbf{x}$, we obtain

$$\mathbf{L_I} = -(\nabla I_x \mathbf{L}_x + \nabla I_y \mathbf{L}_y) \tag{6}$$

where ∇I_x and ∇I_y are the components along x and y of the image gradient ∇I. Note that it is actually the only image processing step necessary to implement the photometric visual servoing method. This approach features many advantages: it does not require any matching or tracking process. Furthermore, since the image measurements are nothing but the pixel intensity, there are no error in the feature extraction process leading to a very precise realization of the task.

III. Principal Component Analysis (PCA)

It is possible to reduce the image high dimensions, by considering Principal Component Analysis (PCA) techniques. PCA is a standard technique which enables to linearly-project high-dimensional samples onto a low-dimensional feature space, that is called eigenspace. Such method has been shown to be very well-suited for object recognition [20], appearance-based tracking [4], keypoint tracking [16]. As stated, in visual servoing, this approach has been initially proposed by [18] and [12][11].

A. Building the eigenspace

Considering PCA for visual servoing requires two steps. The former is an off-line step which consist in building the eigenspace (a set of eigen images) from a large number of arbitrary images of the considered scene. This can be considered as a learning step. The building of the eigenspace consists in following steps:

- acquire M images (with N pixels) and build 1D column vectors that contain all the pixels of the image. We thus have M vector $\mathbf{I}_1, \ldots, \mathbf{I}_M$. We assume that $M \ll N$.
- we then build a $N \times M$ matrix $\mathbf{A}$ whose columns are the normalized image vector $\mathbf{A}_{\bullet i}$:

$$\forall i = 1 \ldots M,\ \mathbf{A}_{\bullet i} = \mathbf{I}_i - \bar{\mathbf{I}},\ \text{with } \bar{\mathbf{I}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{I}_i \tag{7}$$

- Compute the eigenvalues σ_i and their corresponding eigenvectors $\mathbf{U}_i$ of the covariance matrix $\mathbf{C}$ given by

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{A}_{\bullet i}\mathbf{A}_{\bullet i}^{\top} = \mathbf{A}\mathbf{A}^{\top} \tag{8}$$

Note that $\mathbf{C}$ is then a $N \times N$ matrix. Considering a Singular Value Decomposition (SVD) of $\mathbf{A}$ allows to efficiently compute the eigenvectors $\mathbf{U}_{\bullet i}$. Indeed, when decomposing $\mathbf{A}$ such that $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, $\mathbf{U}_{\bullet i}$ is the eigenvector associated to the eigen value σ_i in $\mathbf{\Sigma}$. For simplicity issue, we now denote $\mathbf{U}_i$ the ith eigenvector. This a vector of size N.

- Finally, we can keep only K eigenvectors corresponding to K largest eigenvalues. These vectors create a new orthogonal basis named eigenspace of dimensions K ($K < M$).

B. Image decomposition and reconstruction

Once an eigenspace is built, we have a new orthogonal basis $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_M$ on which we can project each new image and obtain a new and more compact representation of this image. The k-th coordinates of an incoming image (with $k < K$) in this new basis is given by:

$$w_k = \mathbf{U}_k^\top (\mathbf{I} - \bar{\mathbf{I}}) \tag{9}$$

When projecting an incoming image on this basis, the greatest variance comes to lie on the first coordinate w_1, the second greatest variance lies on w_2, etc. When k increases the component w_k becomes less and less significant. It can be noted that the proportion of the variance that each eigenvector represents can be calculated by dividing the corresponding eigenvalue by the sum of all eigenvalues. Following this principle, an image can be represented as linear combination of the eigenimage $\mathbf{U}_i$ and described only by a vector $\mathbf{w} = (w_1, \ldots, w_K)$ with ($K \ll M$) which is a very compact representation. If one wants to reconstruct the image from the vector $\mathbf{w}$, it can be computed as

$$\mathbf{I}_R = \bar{\mathbf{I}} + [\mathbf{U}\mathbf{w}]_K \;\text{ with }\; [\mathbf{U}\mathbf{w}]_K = \sum_{k=1}^{K} w_k \mathbf{U}_k^\top \tag{10}$$

with $K \leq M$. $[\mathbf{U}\mathbf{w}]_K$ is the projection of the image $\mathbf{I}$ onto the subspace defined by the first K basis vectors. $\mathbf{I}_R$ is then approximation of $\mathbf{I}$.

IV. PCA-BASED VISUAL SERVOING

We now present how the PCA can be used within a visual servoing control law.

A. Photometric visual servoing on the reconstructed images

As stated in the previous section the current image $\mathbf{I}(\mathbf{r})$ can be decomposed using equation (9) and then reconstruct using equation (10). The image $[\mathbf{U}\mathbf{w}(\mathbf{r})]$ is an approximation of $\mathbf{I}(\mathbf{r})$ that correspond to the least-square estimate of $\mathbf{w}$ [4]. The coefficients $\mathbf{w}$ are those that minimize $\| [\mathbf{U}\mathbf{w}(\mathbf{r})] - \mathbf{I}(\mathbf{r}) \|^2$. Note that this reconstruction can be achieved using a limited number of eigenvectors K. In that case, and following the methodology presented in [8], [7], the error to be minimized is given by:

$$\mathbf{e}(\mathbf{r}) = \mathbf{I}_R(\mathbf{r}) - \mathbf{I}_R^* = [\mathbf{U}\mathbf{w}(\mathbf{r})]_K - [\mathbf{U}\mathbf{w}^*]_K \tag{11}$$

Within equation (11), $\mathbf{U}$ is constant and only the coefficient $\mathbf{w}(\mathbf{r})$ depends of the camera pose. $[\mathbf{U}\mathbf{w}(\mathbf{r})]$ being an image (we discard subscript K without loss of generality), it can be use as is within a photometric visual servoing process, as defined in section II-B, that minimizes $\mathbf{e}(\mathbf{r})$. The interaction matrix related to $[\mathbf{U}\mathbf{w}(\mathbf{r})]$ is given by $\mathbf{L}_{[\mathbf{U}\mathbf{w}(\mathbf{r})]}$ defined by:

$$\mathbf{L}_{[\mathbf{U}\mathbf{w}(\mathbf{r})]} = -\left(\nabla[\mathbf{U}\mathbf{w}(\mathbf{r})]_x \mathbf{L}_x + \nabla[\mathbf{U}\mathbf{w}(\mathbf{r})]_y \mathbf{L}_y\right) \tag{12}$$

with

$$\begin{aligned} \mathbf{L}_x &= \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \end{pmatrix} \\ \mathbf{L}_y &= \begin{pmatrix} 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \end{aligned} \tag{13}$$

The control law is then given by (see Section II-B):

$$\mathbf{v} = -\lambda \big(\mathbf{L}_{[\mathbf{U}\mathbf{w}(\mathbf{r})]}\big)^+ ([\mathbf{U}\mathbf{w}(\mathbf{r})]_K - [\mathbf{U}\mathbf{w}^*]_K). \tag{14}$$

Such an approach based on reconstructed images allows to remain in a known framework introduced in [8]. Nevertheless, the considered images are built from the K eigenimages that contains the greatest variance. It still contains rich information and mainly discards the details.

B. Visual servoing from the compact eigenspace representation

An alternative to the image reconstruction process presented in the previous paragraph is to directly use the vector $\mathbf{w}$ as the visual feature. This what first was proposed in [12][11]. In these works, the interaction matrix was estimated on-line leading to important approximations and sub-optimal robot motion. In this paper we proposed an analytical formulation of the interaction matrix.

The error to be minimized is then given by:

$$\mathbf{e}(\mathbf{r}) = \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \tag{15}$$

with, for $k = 1..K$:

$$w_k(\mathbf{r}) = \mathbf{U}_k^\top (\mathbf{I}(\mathbf{r}) - \bar{\mathbf{I}}) \tag{16}$$

Having defined the cost function to be minimized, one has to compute the interaction matrix $\mathbf{L}_{w_k} = \frac{\partial w_k}{\partial \mathbf{r}}$ that links the variation of w_k to the camera motion. Within the former equation $\mathbf{U}_k$ and $\bar{\mathbf{I}}$ are constant thus

$$\frac{\partial w_k}{\partial \mathbf{r}} = \frac{\partial \mathbf{U}_k^\top (\mathbf{I}(\mathbf{r}) - \bar{\mathbf{I}})}{\partial \mathbf{r}} = \frac{\partial \mathbf{U}_k^\top \mathbf{I}(\mathbf{r})}{\partial \mathbf{r}} \tag{17}$$

since $\partial \mathbf{U}_k^\top \bar{\mathbf{I}} / \partial \mathbf{r} = 0$ ($\bar{\mathbf{I}}$ being constant), pursuing the derivation of equation (17), we have:

$$\frac{\partial w_k}{\partial \mathbf{r}} = \frac{\partial \mathbf{U}_k^\top}{\partial \mathbf{r}} \mathbf{I}(\mathbf{r}) + \mathbf{U}_k^\top \frac{\partial \mathbf{I}(\mathbf{r})}{\partial \mathbf{r}} \tag{18}$$

Since $\mathbf{U}_k$ is a constant not depending of $\mathbf{r}$ (it has been learnt off-line), this leads to:

$$\mathbf{L}_{w_k} = \mathbf{U}_k^\top \mathbf{L}_{\mathbf{I}}. \tag{19}$$

where $\mathbf{L}_{\mathbf{I}} = \partial \mathbf{I}(\mathbf{r}) / \partial \mathbf{r}$ is given by:

$$\mathbf{L}_I = -\left(\nabla I_x \mathbf{L}_x + \nabla I_y \mathbf{L}_y\right) \tag{20}$$

The complete control law is then given by:

$$\mathbf{v} = -\lambda {\mathbf{L}_{\mathbf{w}}}^+ \big(\mathbf{w}(\mathbf{r}) - \mathbf{w}^*\big). \tag{21}$$

with $\mathbf{L_w} = \mathbf{U}_K^\top \mathbf{L_I}$ where $\mathbf{U}_K$ are the K first columns of $\mathbf{U}$. From a practical point of view we considered a Levenberg-Marquardt-like control law given by:

$$\mathbf{v} = -\lambda \big(\mathbf{H} + \mu\, diag(\mathbf{H})\big)^{-1} \mathbf{L}_\mathbf{w}^\top \big(\mathbf{w}(\mathbf{r}) - \mathbf{w}^*\big). \qquad (22)$$

with $\mathbf{H} = \mathbf{L}_\mathbf{w}^\top \mathbf{L}_\mathbf{w}$ is an approximation of the Hessian. More precisely, each component of the gradient is scaled according to the diagonal of the Hessian, which leads to larger displacements along the direction where the gradient is low. Such a control law has proven its effectiveness in a context of DVS [7], [10]. Note that beside gains λ and μ in equation (22) and, obviously K, no parameters are involved in these experiment. In all the experiment described below, we set $\lambda = 1$ and $\mu = 0.01$. μ decreases by a factor 0.99 at each iteration. Therefore, the control law tends to the classical visual servoing control law that is similar to a Gauss-Newton minimization process (see equation (3)).

With respect to [11] we have there an analytic formulation of the interaction matrix (or image Jacobian) and it is not necessary to estimate it on-line from the image sequence from the estimation of the plane coordinates tangent to the cost function surface leading to a more precise calculation of $\mathbf{L_w}$.

V. Experimental results

Experiments have been carried out both in simulation and on a 6-DOF anthropomorphic robotic arm (a Viper 850 from Adept Company) equipped with a camera mounted on the end-effector. The camera calibration as well as the end-eye calibration have been done in an off-line step. The image processing and the control law computation are performed on a PC equipped with a Dual-core 2.4 Ghz Intel Pentium. The code has been written in C++ using the ViSP library [17]. The time required for an iteration of the VS closed loop is linear wrt. the number of selected coefficients K. In our experiment an iteration corresponds to $31ms$ for $K = 20$ and $73ms$ for $K = 50$ (including image acquisition). We mainly considered two scenes. One is a planar scene (*hollywood*) that also corresponds to the simulation experiments. The second is a 3D castle (*castle3d*) that is 0.2m high (see Figure 1). When dealing with simulation we considered the same calibration parameters as the real system leading to realistic simulations. Simulations have been done thanks to the ViSP simulator [17]. A video of the results can been at https://youtu.be/HncORj8Hpjk.

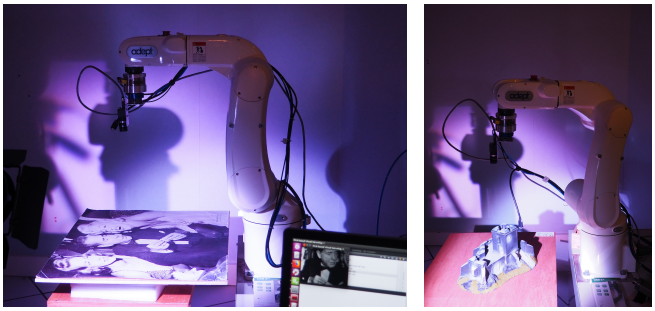

Figure 1. Experimental setup: camera mounted on a Viper 850 from ADEPT and the two considered scenes

A. Construction of the eigenspace

The construction of the eigenspace is done thanks to the acquisition of M image $\mathbf{I}_i, i = 1...M$ of the scene. It is done by just moving the camera handled around the scene. As far as simulation is concerned we generate random camera positions and get the corresponding images. Typically between 2500 and 8000 images are acquired (5000 for the simulations). A SVD decomposition of $\mathbf{A}$ (see equation (7)) is achieved to compute the eigenvectors $\mathbf{U}_k$. Since $\mathbf{A}$ is a large matrix (number of pixel $\times$ number M of images), it takes time (typically 522s when $\mathbf{A}$ is of size 66000×5000). This is done off-line and $\mathbf{U}$ is saved and used for all the experiments. Figure 2 shows the firsts and the last eigenimages for the two scenes *hollywood* and *castle3d*. As expected, the last eigenimage accounts mainly for noise.

The determination of the number K of coefficients is an important problem. As stated in section III-B, each coefficient explains a given percentage of the variance of the data which is directly related to the normalize value of the corresponding eigenvalue (see Figure 2). If we compute the cumulative value of the normalized eigenvalue, one can choose K such that it explains, eg, 75% of the variance which correspond to $K = 20$ on Figure 2). If the M images used to construct the eigenspace are well chosen, less coefficients are necessary to explain the same variance. In practice, we never used more than 50 coefficients.

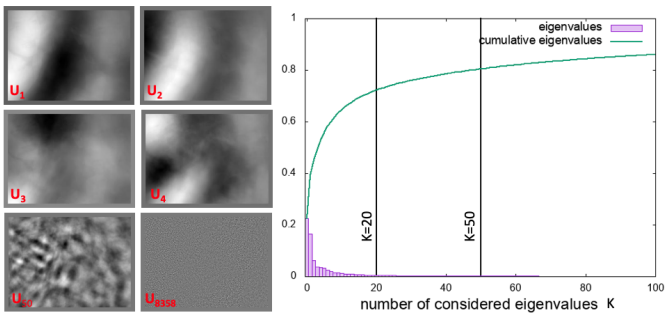

Figure 2. The first four eigenimages $\mathbf{U}_1...\mathbf{U}_4$ along with $\mathbf{U}_{50}$ and the last one $\mathbf{U}_{8358}$ for the *hollywood* scene. As expected, the last eigenimage accounts mainly for noise. Plot on left show the cumulative eigenvalues along with the eigenvalues themselves. One can see that with $K = 20$, we grasp almost 75% of the variance.

B. Simulation results

Simulation results have first been carried out in order to validate the proposed control laws while allowing a fair comparison of different direct visual servoing approaches. We first compare the photometric visual servoing control law [8], [7] with the proposed one. The error between the initial and desired pose is $\mathbf{\Delta r} = (-0.05m, -0.1m, -0.03m, -5^o, 5^o, -15.3^o)$[1]. Photo-

[1]The following notation has been used: $\mathbf{\Delta r} = (\mathbf{t}, \theta\mathbf{u})$, where $\mathbf{t}$ describes the translation part of the homogeneous matrix related to the transformation from the current to the desired frame, while its rotation part is expressed under the form $\theta\mathbf{u}$, where $\mathbf{u}$ represents the unit rotation-axis vector and θ the rotation angle around this axis. This representation is also considered in the plot reporting the positioning error.

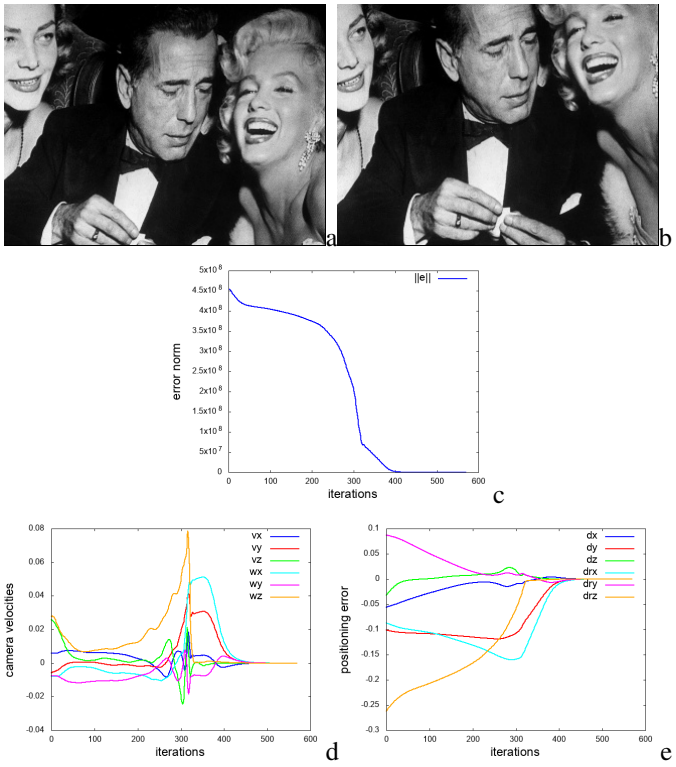

Figure 3. Experiment `exp-photo`: Photometric visual servoing [7]. (a) Initial image $\mathbf{I}$ (b) desired image $\mathbf{I}^*$ (c) error norm $\| \mathbf{I}(\mathbf{r}) - \mathbf{I}^* \|$ (d) camera velocity (in m/s and rad/s) (e) positioning error (in m and rad).

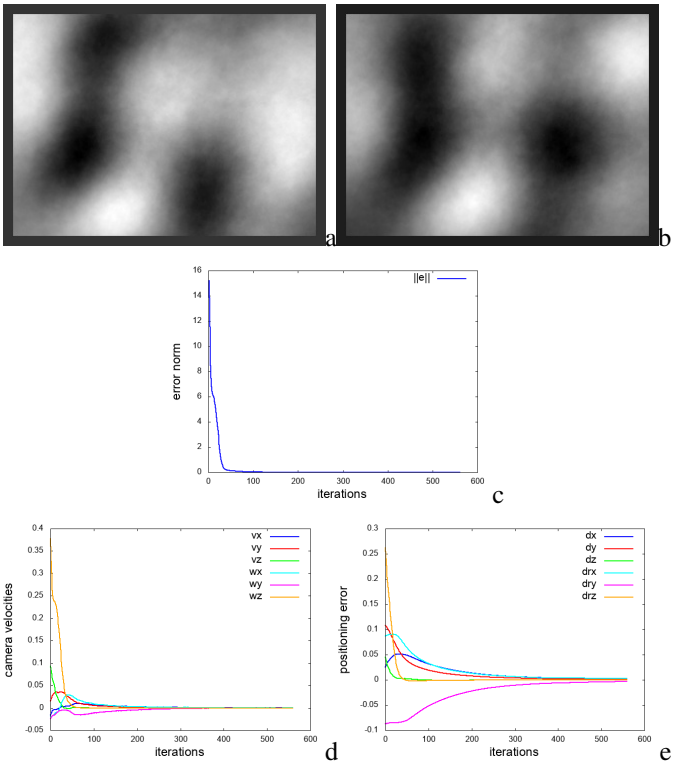

Figure 4. Experiment `exp-photo-rec`: Photometric visual servoing from the reconstructed image $[\mathbf{Uw}(\mathbf{r})]_K$ with $K = 20$. (a) initial reconstructed image $[\mathbf{Uw}(\mathbf{r})]_K$ (b) desired reconstructed image $[\mathbf{Uw}^*]_K$ (c) error norm $\| [\mathbf{Uw}(\mathbf{r})]_\mathbf{K} - [\mathbf{Uw}^*]_\mathbf{K} \|$ (d) camera velocity (in m/s and rad/s) (e) positioning error (in m and rad).

metric visual servoing results, where the cost function to be minimized is given by $\mathbf{e}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) - \mathbf{I}^*$, are shown on Figure 3 (we called it `exp-photo`). The results for the approach presented in section IV-A, where the cost function to be minimized is given by $\mathbf{e}(\mathbf{r}) = \mathbf{I}_R(\mathbf{r}) - \mathbf{I}_R^* = [\mathbf{Uw}(\mathbf{r})]_K - [\mathbf{Uw}^*]_K$ is presented in Figure 4 (this is `exp-photo-rec`). Finally, the proposed subspace approach, where the cost function to be minimized is given by $\mathbf{e}(\mathbf{r}) = \mathbf{w}(\mathbf{r}) - \mathbf{w}^*$ is presented in Figure 6 for $K = 6$ (this is `exp-w6`) and in Figure 7 for $K = 20$ (this is `exp-w20`). For all the experiment we show the norm of the cost function, the camera velocity (in m/s and rad/s) and the positioning error (in m and rad).

When considering pure photometric visual servoing (Experiment `exp-photo`, Figure 3), the photometric cost function is highly non-linear which explain the perturbation in the velocity plots. Nevertheless, the visual servoing converges thanks to the redundancy of the considered information. Such experiment are in line with previous experiments (eg [8]). Dealing with photometric visual servoing from the reconstructed images (see Section IV-A and Experiment `exp-photo-rec`, Figure 4) the camera behavior is far better and the camera trajectory is closer from a geodesic. The camera velocities (Figure 4.d) ensures a clear monotonous decrease of the cost function norm (Figure 4.c) and of the positioning error (Figure 4.e). This is due to the fact that when considering only 20 coefficients to reconstruct the images, it smoothes the images (see Figure 4(a-b)) and reduces the non-linearties in the cost function and thus increases the convergence area (a similar

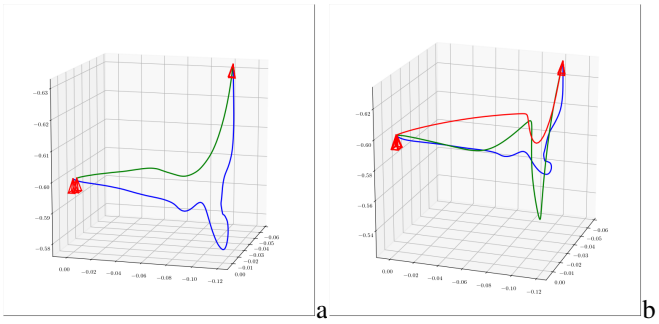

Figure 5. 3D camera trajectories (a) blue: `exp-photo` ; green: `exp-photo-rec` (b) blue: `exp-photo-rec` ; green: `exp-w6` ; red: `exp-w20`

conclusion was reached by [9] using photometric Gaussian Mixtures as visual features). Comparison between camera trajectories for `exp-photo` and `exp-photo-rec` is given in Figure 5.a.

We then consider directly the coefficients of the compact representation. Let us recall that this is a size K vector. Let us point out that $K = 6$, is the minimum number of coefficients necessary to have a full rank interaction matrix and thus to control the 6 DoF of the camera. We first consider two experiments the former with $K = 6$ and the latter with $K = 20$. In both cases, the visual servoing control law converges. Nevertheless, our experiments show that considering

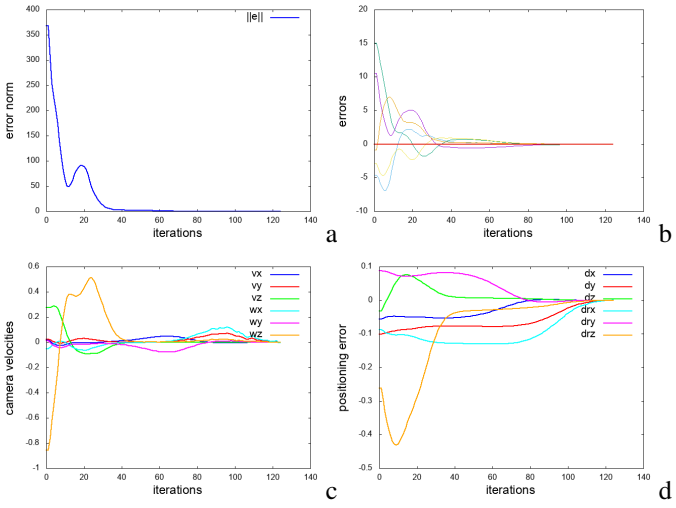

Figure 6. Experiment `exp-w6`: Positioning task using compact eigenspace representation with $K = 6$ (a) error norm $\| \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \|$ (b) errors $w_i(\mathbf{r}) - w_i^*, i = 1..6$ (c) camera velocity (in m/s and rad/s) (d) positioning error (in m and rad)

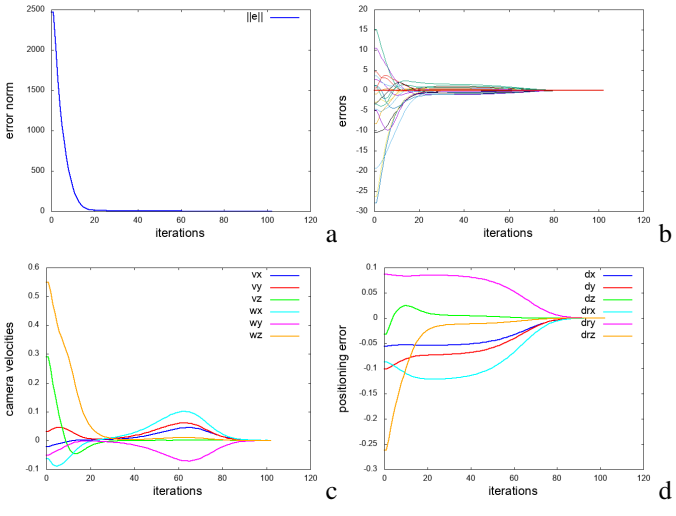

Figure 7. Experiment `exp-w20`: Positioning task using compact eigenspace representation with $K = 20$ (a) error norm $\| \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \|$ (b) errors $w_i(\mathbf{r}) - w_i^*, i = 1..20$ (c) camera velocity (in m/s and rad/s) (d) positioning error (in m and rad).

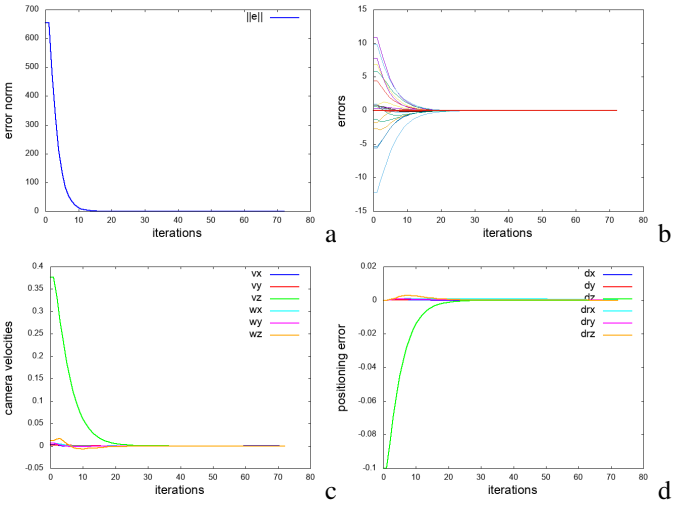

Figure 8. Positioning task using compact eigenspace representation with $K = 20$ for a pure translation motion $\mathbf{\Delta r} = (0, 0, 0.1m, 0, 0, 0)$ (a) error norm $\| \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \|$ (b) errors $w_i(\mathbf{r}) - w_i^*, i = 1..20$ (c) camera velocity (in m/s and rad/s) (d) positioning error (in m and rad)

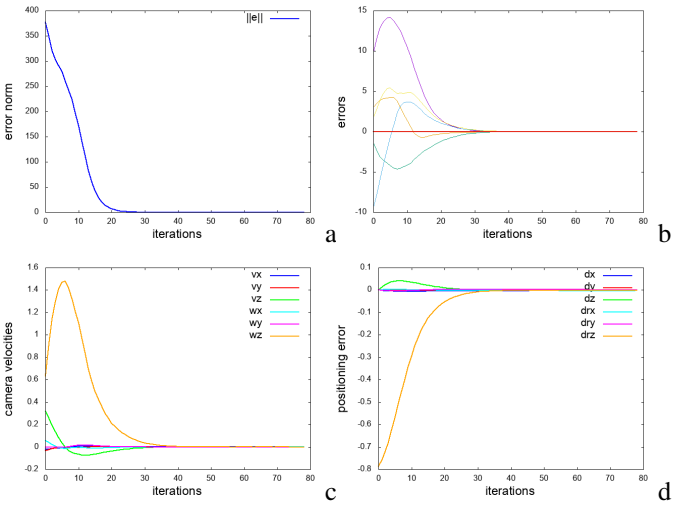

Figure 9. Positioning task using compact eigenspace representation with $K = 6$ for a pure rotation motion around Z axis. $\mathbf{\Delta r} = (0, 0, 0, 0, 0, 0, 45^o)$ (a) error norm $\| \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \|$ (b) errors $w_i(\mathbf{r}) - w_i^*, i = 1..20$ (c) camera velocity (in m/s and rad/s) (d) positioning error (in m and rad)

at least 15 coefficients leads to better behavior and a better monotonous decrease of the cost function norm (see 7.a vs 6.a). The convergence is also much faster than with the two previous photometric-based methods (140 vs 600 iterations). The small overshoot that can be observed in the camera velocities between iteration 35 and 80 (Figure 7) is due to decay of the parameters μ in equation (22). When μ is high (typically 0.01), the control law produces fast motion along the direction where the gradient is low. It then reach very quickly a position where the cost function is relatively low and where the error in the image space $(\mathbf{I} - \mathbf{I}^*)$ is very small although the error in the cartesian space is still important. Then when μ decreases, the control law tends to the classic one which is more precise when the cost function decreases (thanks to a better estimation of the interaction matrix). This explains the temporary increase of the velocity. The 3D trajectories are show on Figure 5.b.

We also show the results obtained for two specific motions: a pure 0.10 m translation along the optical axis (see Figure 8) and a pure 45^o rotation around the optical axis (see Figure 9). In both cases, the control law generates a motion along the desired axis. This tends to show that, at least in these cases, there are almost no coupling between translational and rotational camera motions leading to very satisfactory camera trajectories [5]. Dealing with the rotation which is quite large, a pure photometric approach fails and a classical IBVS method based on point coordinates would features a significant motion along the z axis (camera retreat). This motion exists here (see Figure 9.d) but is almost insignificant.

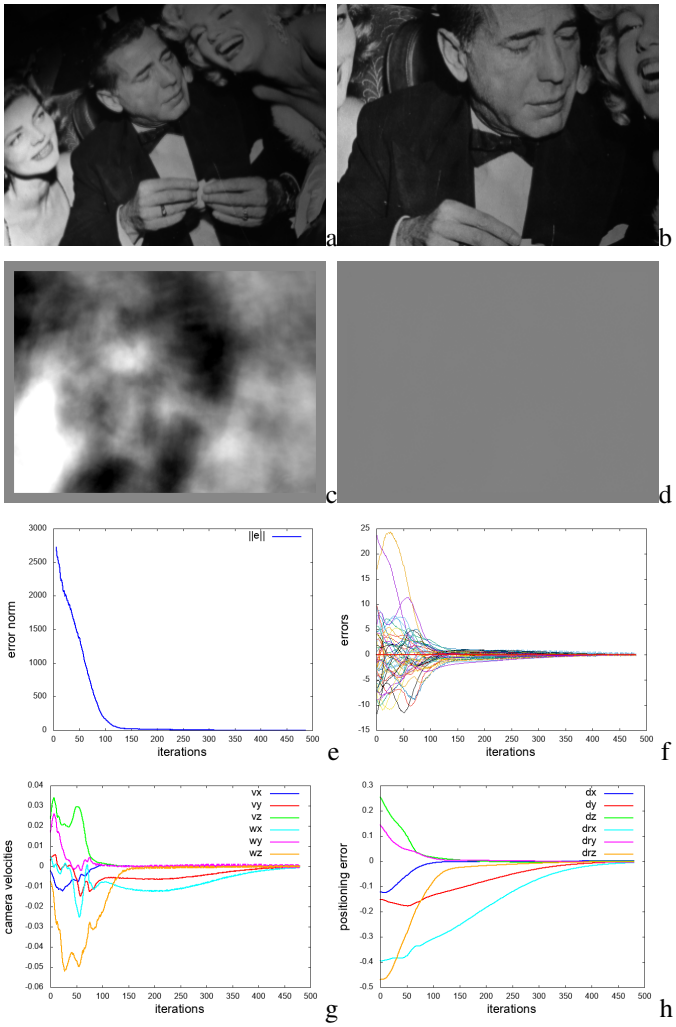

Figure 10. Experiment with real planar scene (a) initial image acquired by the camera $\mathbf{I}(\mathbf{r})$, (b) image $\mathbf{I}^*$ acquired from the desired position, (c,d) error $[\mathbf{Uw}(\mathbf{r})]_{50} - [\mathbf{Uw}^*]_{50}$ between reconstructed image for initial and desired position (a,b,c,d) are used for visualization but are not used in the algorithm. Only the error $\mathbf{w}(\mathbf{r}) - \mathbf{w}^*$ plotted in (f) is considered, (e) $\parallel \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \parallel$ (g) camera velocity (in m/s and rad/s) (h) positioning error (in m and rad).

C. Experimental results on a 6 DoF robot

We report here three experiments carried out on the Viper 850 robot. The first two ones consider the *hollywood* scene and the later the *castle3d* scene.

a) 6 DoF positioning task: This experiment reports a 6 DoF positioning task with respect to a planar scene with $K = 50$. The displacement to be achieved is $\mathbf{\Delta r} = (0.04m, 0.27m, 0.04m, 22.3^o, 8^o, 26.3^o)$. Let us point out that the transformation between the initial and desired poses (and particularly the rotation around the x and z axes) is very large and makes this experiment very challenging. This is also illustrated by the initial and desired images depicted in Figure 10(a-b). The norm of the cost function $\parallel \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \parallel$ decreases monotonously (Figure 10.e). The decrease in errors (Figure 10.d) is also highly satisfactory considering the fact that only the interaction matrix at the desired position and an approximated depth were employed. One can see on Figure 10.h that the control law tends to minimize rapidly the errors accounting for the z translation and z rotation (at iteration 200, the positioning error for these axis are almost null). The final error is $\mathbf{\Delta r} = (0.0012m, 0.0023m, 0.00001m, -0.32^o, 0.19^o, 0.06^o)$ which show the accuracy of the proposed approach. Although we consider here $K = 50$, the camera converges for smaller value of K. Figure 11 show the camera trajectories for $K = 20$ (blue) and $K = 50$ (green). The camera trajectories are close to a geodesic.

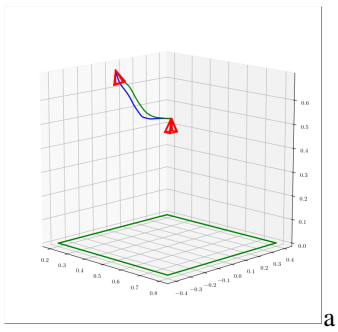

Figure 11. 3D camera trajectories for experiment (blue: $K = 20$; green: $K = 50$).

b) Dealing with partial occlusions: In the next experiment we add large occlusions by two (non planar) objects (see Figure 12). We still consider the eigenspace learnt without occlusions (the same as for the previous experiment). We consider $K = 50$. We still consider large displacement in $\mathbf{\Delta r} = (-0.18m, -0.07m, 0.04m, -18^o, 14.3^o, 28.6^o)$. The final error is $\mathbf{\Delta r}$ = (-0.0024m, -0.0017m, 0.0003m, 0.25^o, 0.25^o, 0.05^o) which shows that the occlusions and the non-planar scene does not affect the precision of the positioning task. Nevertheless, it can be noted that positioning errors decrease less monotonously due to larger modelling errors in the calculation of the interaction matrix. This experiment demonstrates the robustness of the control law to large displacements, partial occlusions, and modelling errors.

c) Dealing with a 3D scene: In order to demonstrate further the robustness of the approach, we propose to consider in this experiment a large 3D object (Figure 13). For this last experiment, we consider $\mathbf{\Delta r} = (0.15m, 0.21m, -0.13m, 16.6^o, 13.7^o, 30.9^o)$. We only consider $K = 20$ coefficients in vectors $\mathbf{w}$ and $\mathbf{w}^*$. Although, in this case, we trained a new eigenspace, it is important to note that depth is still assumed to be constant ($Z = 0.6m$), whereas the object high is almost 0.2m, leading to modelling errors in the interaction matrix. These modelling errors, that could be overcome with a RGB-D camera, affect the behavior of the control law as can be seen on the error and velocities plots. Nevertheless, the system converges and the final error is $\mathbf{\Delta r} = (-0.0007m, -0.0008m, 0.0007m, 0.01^o, 0.01^o, 0.01^o)$ (we increase the number of iteration leading to a better precision that for the two previous experiments).

VI. Conclusion

In this paper we demonstrated that direct visual servoing techniques can take advantage from the projection of the

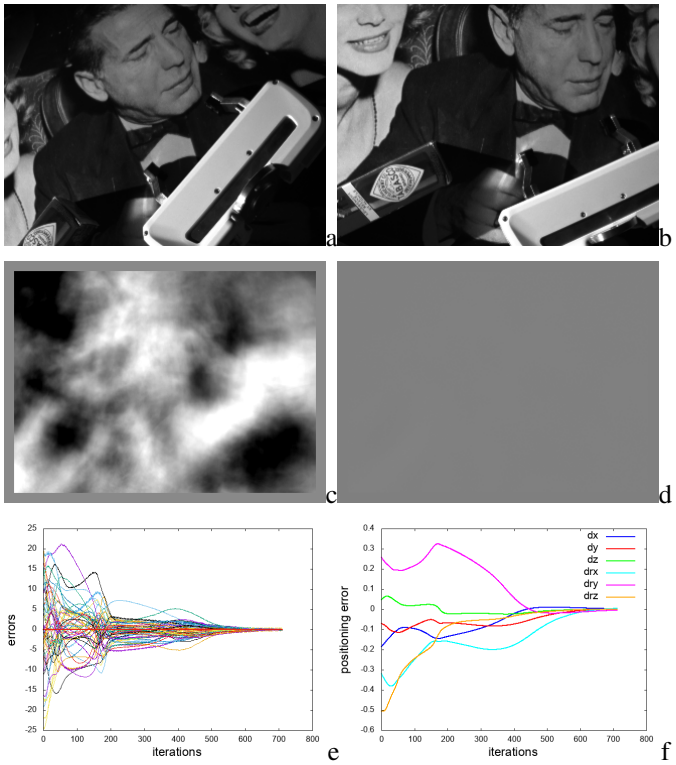

Figure 12. Experiment real scene with partial occlusions (a) initial image acquired by the camera $\mathbf{I}(\mathbf{r})$, (b) image $\mathbf{I}^*$ acquired from the desired position, (c-d) error $[\mathbf{U}\mathbf{w}(\mathbf{r})]_{50} - [\mathbf{U}\mathbf{w}^*]_{50}$ between reconstructed image for initial and desired position used for visualization but not used in the algorithm. (e) errors $\mathbf{w}(\mathbf{r}) - \mathbf{w}^*$ (f) positioning error (in m and rad).

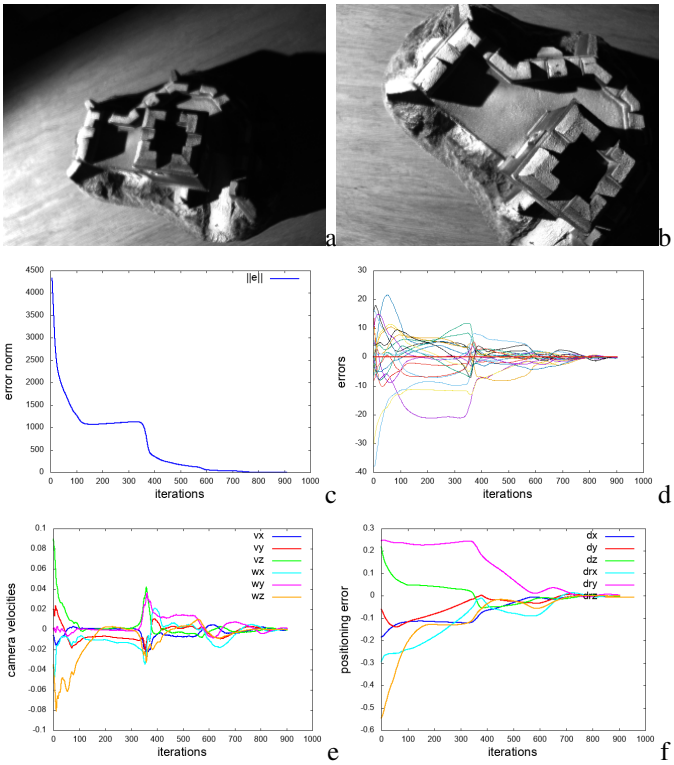

Figure 13. Experiment with a large 3D scene (a) initial image acquired by the camera $\mathbf{I}(\mathbf{r})$, (b) image $\mathbf{I}^*$ acquired from the desired position, (c) $\| \mathbf{w}(\mathbf{r}) - \mathbf{w}^* \|$ (d) errors $\mathbf{w}(\mathbf{r}) - \mathbf{w}^*$ (e) camera velocity (in m/s and rad/s) (f) positioning error (in m and rad).

image on a new basis. It was also shown that the interaction matrix related to these new coordinates can be explicitly and analytically calculated. Results show the effectiveness of this approach on various examples. Future works will be devoted to study other projection techniques (such as Linear Discriminant Analysis).

REFERENCES

[1] M. Bakthavatchalam, O. Tahri, and F. Chaumette. A Direct Dense Visual Servoing Approach using Photometric Moments. *IEEE Trans. on Robotics*, 34(5):1226–1239, October 2018.

[2] Q. Bateux and E. Marchand. Histograms-based visual servoing. *IEEE Robotics and Automation Letters*, 2(1):80–87, January 2017.

[3] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke. Training deep neural networks for visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'18*, pages 3307–3314, Brisbane, Australia, May 2018.

[4] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. Journal of Computer Vision*, 26(1):63–84, January 1998.

[5] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D.J. Kriegman, G. Hager, and A.S. Morse, editors, *The confluence of vision and control*, Lecture Notes in control and information sciences, No 237, pages 67–78. Springer, June 1997.

[6] F. Chaumette and S. Hutchinson. Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.

[7] C. Collewet and E. Marchand. Photometric visual servoing. *IEEE Trans. on Robotics*, 27(4):828–834, August 2011.

[8] C. Collewet, E. Marchand, and F. Chaumette. Visual servoing set free from image processing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, pages 81–86, Pasadena, CA, May 2008.

[9] N. Crombez, E.M. Mouaddib, and G. Caron. Photometric Gaussian mixtures based visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'15*, pages 5486–5491, Hamburg, Germany, September 2015.

[10] A. Dame and E. Marchand. Mutual information-based visual servoing. *IEEE Trans. on Robotics*, 27(5):958–969, October 2011.

[11] K. Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *Int. Journal of Computer Vision*, 37(1):7–20, June 2000.

[12] K. Deguchi and T. Noguchi. Visual servoing using eigenspace method and dynamic calculation of interaction matrices. In *IAPR Int Conf on Pattern Recognition*, pages 302–306, Vienna, Austria, August 1996.

[13] L.-A. Duflot, R. Reisenhofer, B. Tamadazte, N. Andreff, and A. Krupa. Wavelet and Shearlet-based Image Representations for Visual Servoing. *The Int. Journal of Robotics Research*, 38(4):422–450, April 2019.

[14] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[15] V. Kallem, M. Dewan, J.P. Swensen, G.D. Hager, and N.J. Cowan. Kernel-based visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07*, pages 1975–1980, San Diego, USA, October 2007.

[16] Y. Ke and R. Sukthankar. PCA-SIFT a more distinctive representation for local image descriptor. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 506–513, Washington, DC, July 2004.

[17] E. Marchand, F. Spindler, and F. Chaumette. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52, December 2005.

[18] S.K. Nayar, S.A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Trans. on Robotics*, 12(5):750 – 758, October 1996.

[19] M. Ourak, T. Brahim, O. Lehmann, and N. Andreff. Direct visual servoing using wavelet coefficients. *IEEE/ASME Transactions on Mechatronics*, 2019.

[20] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 586–591, Maui, Hawaii, 1991.